POWERPLAY TECH

Team Members

Daniel Badipehter – Scrum Master/Project Lead

Edinam Foli – QA and Testing Lead

Alexandra Hawa Mahama - QA and Testing

William Bokuro - DevOps / CI-CD Lead

Prince Kwame Amoah - **DevOps / CI-CD**

Cosmas Awupuri Ayikem – Documentation & Demo Lead

Project Title: Eagles Academy Website

Client: Eagles Academy and Adults Education – Kyeremankomah, Kintampo North
Contact: 0553964007

---

1. Introduction & System Intent

Client Overview

Eagles Academy and Adults Education is a private educational institution located in Kyeremankomah, Kintampo North, Ghana. The school serves both children and adults, offering academic instruction and vocational training. Eagles Academy and Adults Education is making great community impact; however it lacks a strong digital presence, which limits its accessibility by students, parents and other stakeholders.

Problem Statement

As it stands, Eagles Academy does not operate a centralized online platform. Parents, teachers, and prospective students face difficulties accessing essential information about the school, such as academic programs, events, enrollment procedures, and contact details. This absence of a reliable website has weakened the institution's visibility, created inefficiencies in communication, and slowed down the enrollment process. Without a structured digital system, the academy risks losing potential learners to better-publicized competitors.

System Vision

To design and implement a modern and responsive website for Eagles Academy that centralizes academic information, events, enrollment, and media resources into one accessible and user-friendly platform.

User Personas

- **Primary Users:** Parents, teachers, and prospective students who require timely access to academic information, enrollment processes, and updates on school activities.
- **Secondary Users:** New visitors, potential partners, and the wider community seeking information about the school.
- **Context of Use:** Parents and students will primarily access the website via mobile and desktop devices, while administrators will update and manage the platform's content.

2. Requirements & Functionality

Functional Requirements (FR)

- **FR-1:** Users can view school information (about, contact, location).

- **FR-2:** Users can access and download the academic calendar and curriculum.
- **FR-3:** Users can view latest news and events.
- **FR-4:** Users can browse a gallery with filter options (by year, event).
- **FR-5:** Users can fill in and submit an online enrollment form.
- **FR-6:** Users can select a date for entrance examinations during enrollment.
- **FR-7:** Search functionality to quickly locate information or media.
- **FR-8:** Responsive navigation bar and smooth page transitions.

Non-Functional Requirements

- The website must be responsive across desktop, tablet, and mobile.
- Pages must load within three seconds under normal conditions.
- Secure form submissions using HTTPS and validation.
- Accessibility compliance with assistive technologies.
- Uptime of at least 99%.
- Modular and maintainable codebase.

MoSCoW Prioritisation

Priority	**Requirements**
Must Have	FR-1, FR-2, FR-3, FR-5, FR-8
Should Have	FR-4, FR-6, FR-7
Could Have	Event reminder notifications, alumni section
Won’t Have	Full student portal (future version)

Preliminary Use-Case Table

Use-Case (UC)	**Brief**	**Actors**
UC-1	Submit online enrollment form	Prospective student, Parent
UC-2	View/download academic calendar	Parent, Teacher, Student

Use-Case (UC)	**Brief**	**Actors**
UC-3	Browse news/events	Parent, Visitor
UC-4	Search gallery by event/year	Visitor, Parent

3. Architecture & Components

System Decomposition

The proposed solution will adopt three-tier architecture:

- **Presentation Layer:** A responsive web user interface developed with HTML, CSS, and JavaScript.
- **Application Layer:** A backend application (PHP) to handle enrollment requests, events, and content management.
- **Database Layer:** A SQL database to store enrollment records, event details, and gallery metadata.

This decomposition ensures

1. **Scalability**
    - By isolating the presentation layer from the business logic and database, the system can scale each tier independently.
    - For example, if user traffic increases significantly, additional web servers can be added to the presentation layer without altering the application logic or database structure.
    - Similarly, the database layer can be optimized or replicated independently to handle larger volumes of enrollment data.
    - This modular separation enables both vertical scalability (improving capacity within a layer) and horizontal scalability (adding more servers), which is essential for future growth.
2. **Maintainability**
    - Each layer has a distinct responsibility: the presentation layer manages the user interface, the application layer manages logic, and the database layer manages data.

- This separation ensures that modifications in one layer do not disrupt others. For instance, the website design can be updated in the presentation layer without affecting the business logic in the application layer.
- The modular structure makes debugging and testing more straightforward, as issues can be isolated within a specific layer.
- Maintainability is also enhanced by enabling parallel development; different team members can work on different layers simultaneously.

3. **Security**
   - Sensitive data is safeguarded at the database layer, which is isolated from direct user interaction. Only the application layer can interact with the database, reducing the risk of unauthorized access.
   - Role-based access controls can be implemented in the application layer to ensure that only authorized staff can manage enrollment records or update school events.
   - The separation of layers also supports secure communication protocols (e.g., HTTPS between client and server) and parameterized queries at the database level to prevent SQL injection attacks.
   - By minimizing direct exposure of the database and centralizing security checks in the application layer, the system reduces attack surfaces and enhances overall resilience.

Primary Models

Data Flow Diagram (DFD)

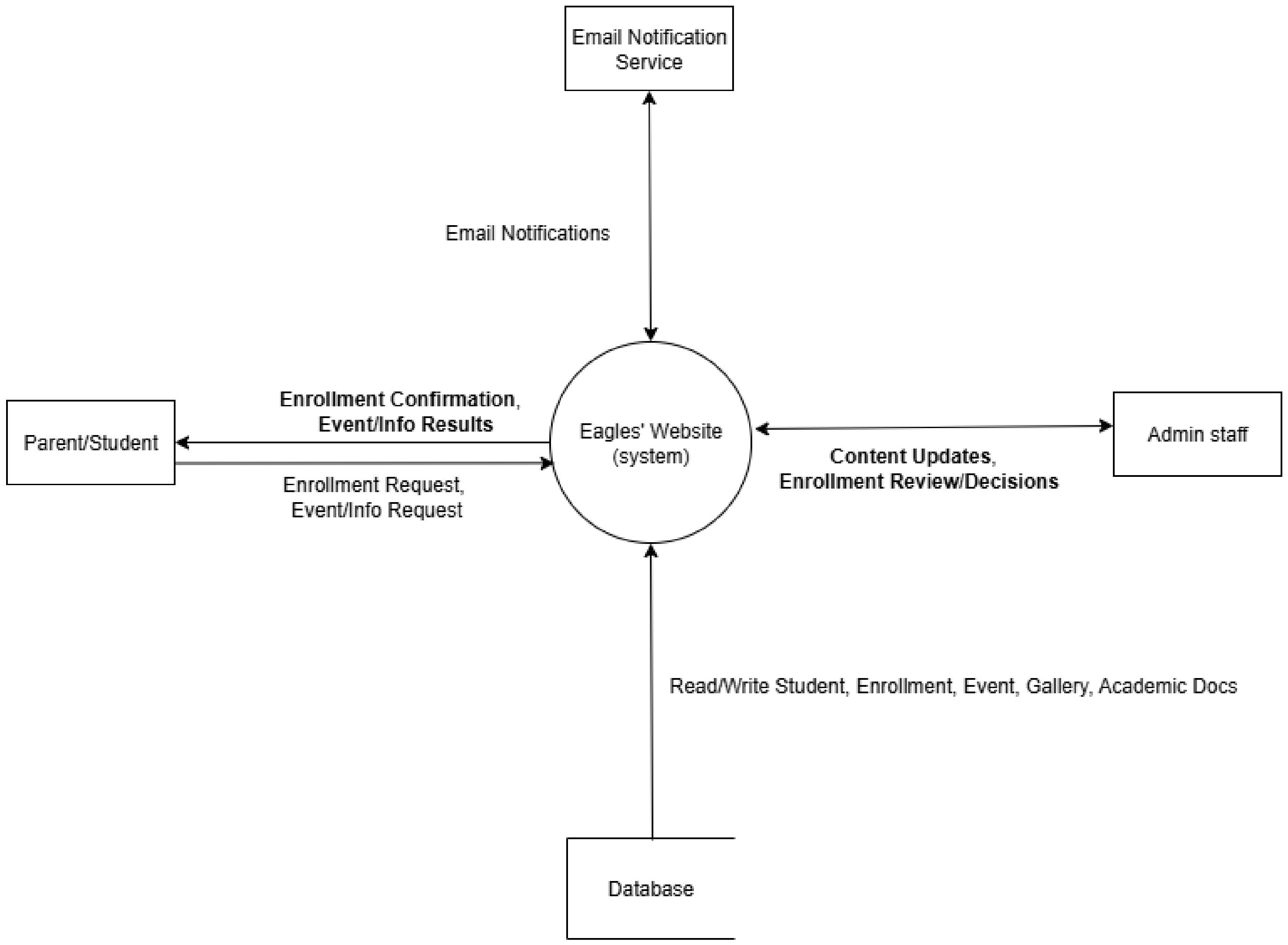

Email Notification Service
Email Notifications
Enrollment Confirmation, Event/Info Results
Parent/Student
Eagles' Website (system)
Admin staff
Enrollment Request, Event/Info Request
Content Updates, Enrollment Review/Decisions
Read/Write Student, Enrollment, Event, Gallery, Academic Docs
Database

Figure 1: Data Flow Diagram

Entity Relationship Diagram (ERD)

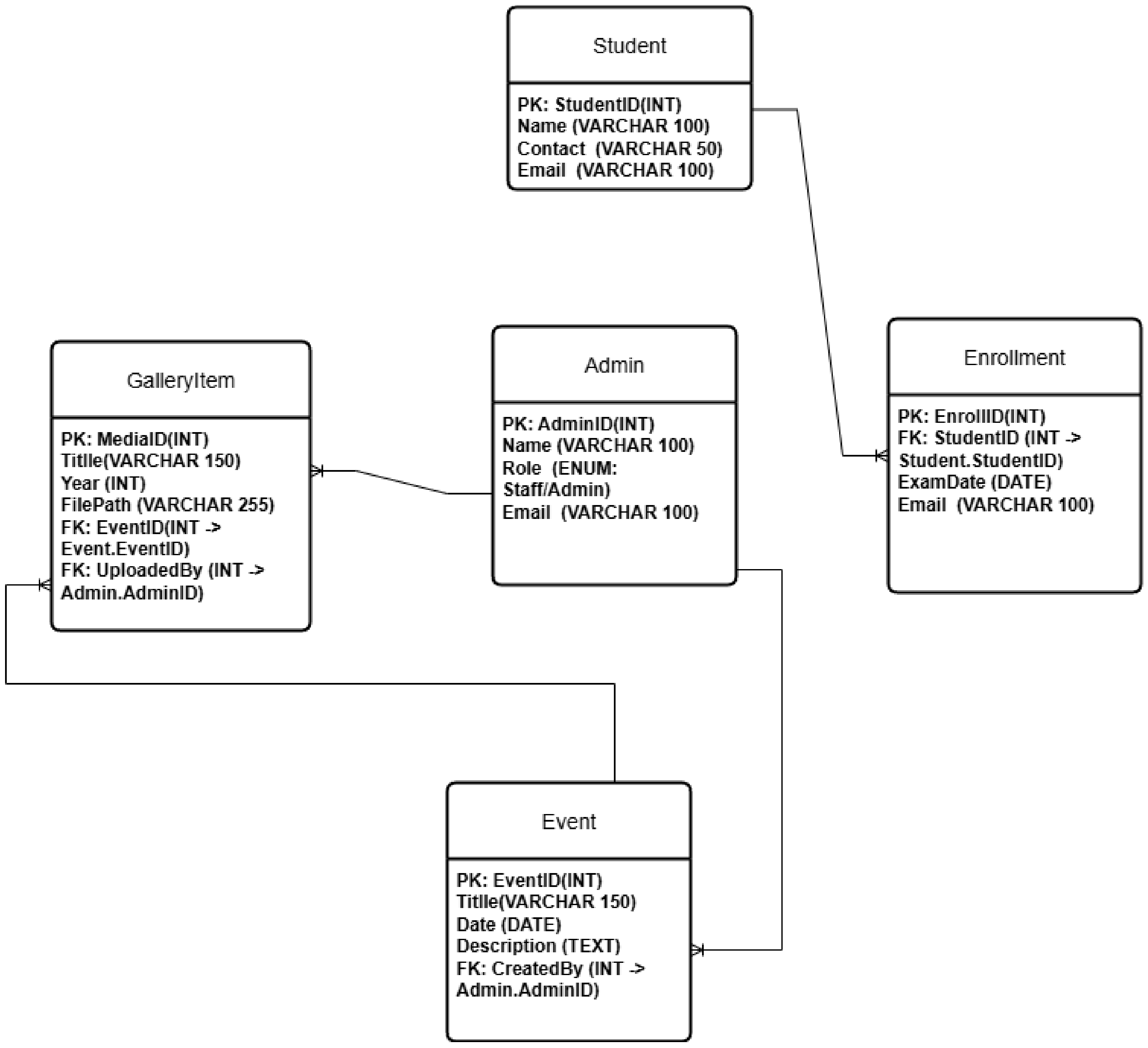


Figure 2: Entity Relationship Diagram

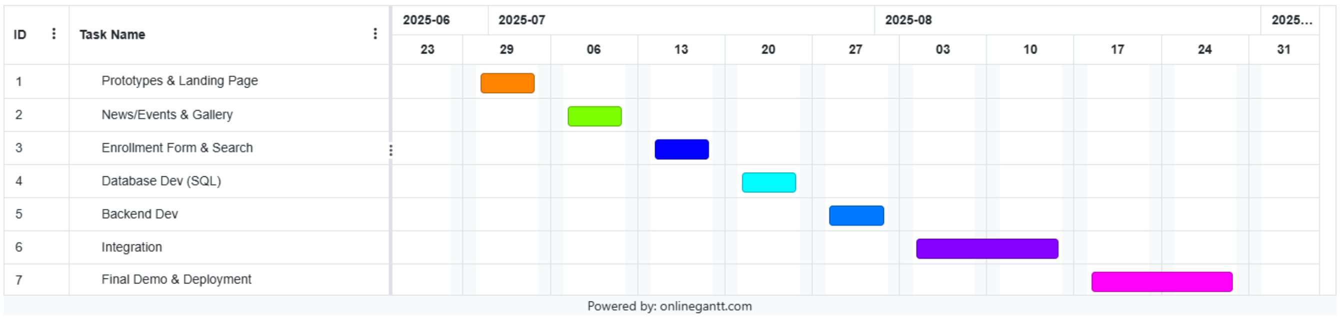
4. Scope, Deliverables & Plan

Deliverable List

- **Sprint 1:** Figma prototypes, ERD design, and static landing page.
- **Sprint 2:** News/events page, gallery module, and academic information pages.
- **Sprint 3:** Enrollment form with exam date selector and search functionality.
- **Sprint 4:** Database Dev (SQL)
- **Sprint 5:** Backend Dev
- **Sprint 6:** Integration
- **Final Demo:** Fully responsive, tested, and deployed website.

Timeline

A mini-Gantt chart will track progress across the three sprints, showing milestones for prototypes, module development, and integration.



ID	Task Name
1	Prototypes & Landing Page
2	News/Events & Gallery
3	Enrollment Form & Search
4	Database Dev (SQL)
5	Backend Dev
6	Integration
7	Final Demo & Deployment

Figure 3: Mini Gantt Chart

Definition of Done

A feature is considered complete when:

- It is fully functional,
- Tested through unit and usability checks,
- Integrated with existing modules,
- Demonstrated to and approved by the client.

5. Communication, Visibility & Risk

Client Touch-points

The project team will engage the client through:

- Weekly face-to-face meetings and progress presentations.

Team Communication

- Daily meetings on weekdays.
- GitHub project board for visibility and tracking.
- WhatsApp channel for urgent updates.

Risk Log

Risk	Probability	Impact	Mitigation
Deadline slippage	Medium	High	Weekly checks and sprint planning
Data loss during development	Low	High	Use GitHub version control and backups
Team skill gaps in PHP/SQL	Medium	Medium	Pair programming and training sessions

6. Development Process & Compliance

The team will adopt **Scrum methodology**, which facilitates iterative development, regular feedback, and adaptability (Pressman, 2019). DevSecOps guardrails will include branch protection, continuous integration pipelines, and secret scanning to avoid exposing sensitive data.

Generative AI tools will be used sparingly for visualization and documentation structuring. However, all code and deliverables will be developed, reviewed, and validated by the team to ensure accountability.

References

Pressman, R. S. (2019). *Software engineering: A practitioner’s approach* (9th ed.). McGraw-Hill.